

Τμήμα Μηχανικών Η/Υ, Τηλεπικοινωνιών και Δικτύων
Πανεπιστημίου Θεσσαλίας



Διπλωματική εργασία

«Σύστημα λήψης ιατρικών
αποφάσεων
| *Medical decision support system*»

Κούλελη Μαρίνα

Ιούλιος 2012, Βόλος

Επιβλέπουσα: κα Δασκαλοπούλου Ασπασία
Επίκουρη καθηγήτρια Π.Θ.
Συνεπιβλέπων: κος Ακρίτας Αλκιβιάδης
Αναπληρωτής καθηγητής Π.Θ.

Ευχαριστίες

Με το πέρας της διπλωματικής μου εργασίας καθώς και την ολοκλήρωση των σπουδών μου, θα ήθελα πρωτίστως να ευχαριστήσω την κυρία Ασπασία Δασκαλοπούλου για την ευκαιρία που μου έδωσε να διεκπεραιώσω αυτή τη διπλωματική εργασία καθώς και τη βοήθειά της πάνω σε αυτή. Ιδιαίτερα ευχαριστώ τους φίλους μου για την υποστήριξή τους όλα αυτά τα χρόνια . Τέλος ένα μεγάλο ευχαριστώ στους γονείς μου για την στήριξη, τη βοήθεια και την υπομονή που έδειξαν κατά τη διάρκεια των σπουδών μου.

Περίληψη

Ένα σύστημα λήψης ιατρικών αποφάσεων είναι ένα σύστημα βασισμένο σε υπολογιστή, με σκοπό την εξαγωγή συμπερασμάτων που βοηθούν στη λήψη ιατρικής απόφασης. Το σύστημα αποτελείται από μια βάση γνώσης, μια μηχανή συμπερασμού και μια διεπαφή με το χρήστη. Στο συγκεκριμένο σύστημα λήψης ιατρικών αποφάσεων που έχουμε υλοποιήσει, μπορούμε εισάγοντας συμπτώματα να καταλήξουμε σε συμπεράσματα όπως ποια είναι η πιθανή ασθένεια δεδομένων των συμπτωμάτων. Αυτό γίνεται με την εκ των προτέρων εισαγωγή κανόνων στη βάση γνώσης, οι οποίοι σχετίζουν τα συμπτώματα με τις ασθένειες. Στη συνέχεια η εφαρμογή δίνει τη δυνατότητα στο χρήστη να εισάγει τα δεδομένα (συμπτώματα) που επιθυμεί και να πάρει τα ανάλογα αποτελέσματα. Παρόμοια συστήματα υπάρχουν ήδη και χρησιμοποιούνται σε διάφορους τομείς της ιατρικής.

ΚΕΦΑΛΑΙΟ 1: ΕΙΣΑΓΩΓΗ	5
ΣΥΣΤΗΜΑΤΑ ΥΠΟΣΤΗΡΙΞΗΣ ΛΗΨΗΣ ΑΠΟΦΑΣΕΩΝ	5
ΠΕΡΙΓΡΑΦΗ	5
ΙΣΤΟΡΙΑ	6
ΕΦΑΡΜΟΓΕΣ	7
ΣΥΣΤΗΜΑΤΑ ΛΗΨΗΣ ΙΑΤΡΙΚΩΝ ΑΠΟΦΑΣΕΩΝ	7
ΠΕΡΙΓΡΑΦΗ	7
ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΚΑΙ ΜΕΘΟΔΟΛΟΓΙΕΣ	8
ΠΑΡΑΔΕΙΓΜΑΤΑ ΣΥΣΤΗΜΑΤΩΝ ΥΠΟΣΤΗΡΙΞΗΣ ΛΗΨΗΣ ΙΑΤΡΙΚΩΝ ΑΠΟΦΑΣΕΩΝ	9
ΚΕΦΑΛΑΙΟ 2: ΠΕΡΙΓΡΑΦΗ ΤΗΣ ΕΦΑΡΜΟΓΗΣ	12
ΚΑΝΟΝΕΣ	12
ΠΑΡΑΔΕΙΓΜΑ	12
ΚΕΦΑΛΑΙΟ 3: ΥΛΟΠΟΙΗΣΗ ΤΟΥ ΠΡΟΓΡΑΜΜΑΤΟΣ	19
ΤΟ ΠΡΟΓΡΑΜΜΑ CLIPS	19
ΚΑΝΟΝΕΣ ΚΑΙ ΓΕΓΟΝΟΤΑ	19
ΓΙΑ ΝΑ ΕΙΝΑΙ ΑΛΗΘΗΣ Ο ΚΑΝΟΝΑΣ ΠΡΕΠΕΙ ΝΑ ΙΣΧΥΟΥΝ ΚΑΙ ΤΑ ΔΥΟ ΑΥΤΑ ΓΕΓΟΝΟΤΑ.	20
ΠΡΟΤΕΡΑΙΟΤΗΤΑ ΚΑΝΟΝΩΝ	20
ΑΛΛΗΛΕΠΙΔΡΑΣΗ ΜΕ ΤΟ ΧΡΗΣΤΗ	22
ΕΝΕΡΓΟΠΟΙΗΣΗ ΤΩΝ ΚΑΝΟΝΩΝ	24
ΈΝΑ ΣΤΙΓΜΙΟΤΥΠΟ ΕΚΤΕΛΕΣΗΣ ΤΟΥ ΠΡΟΓΡΑΜΜΑΤΟΣ	24
ΚΕΦΑΛΑΙΟ 4: ΜΕΛΛΟΝΤΙΚΕΣ ΕΠΕΚΤΑΣΕΙΣ	43
ΠΡΟΣΘΗΚΕΣ ΣΤΟ ΥΠΑΡΧΟΝ ΠΡΟΓΡΑΜΜΑ	43
ΕΠΙΠΛΕΟΝ ΣΥΝΤΕΛΕΣΤΕΣ ΒΕΒΑΙΟΤΗΤΑΣ	43
ΕΠΙΠΛΕΟΝ ΜΕΤΡΗΣΕΙΣ ΚΑΙ ΔΕΙΚΤΕΣ	45
ΕΞΕΛΙΞΗ ΤΟΥ ΠΡΟΓΡΑΜΜΑΤΟΣ	46
ΚΕΦΑΛΑΙΟ 5: ΒΙΒΛΙΟΓΡΑΦΙΑ	47

Κεφάλαιο 1: Εισαγωγή

Συστήματα υποστήριξης λήψης αποφάσεων

Περιγραφή

Τα συστήματα υποστήριξης λήψης αποφάσεων είναι συστήματα βασισμένα σε υπολογιστές τα οποία βοηθούν το χρήστη να πάρει μια απόφαση και συνήθως υποστηρίζουν τη λήψη ημιδομημένων ή αδόμητων αποφάσεων. Συγκεκριμένα, περιέχουν μία βάση γνώσης η οποία αποτελείται από κανόνες, αλληλεπιδρούν με το χρήστη και του παρέχουν ένα σύνολο πιθανών λύσεων. Τα συστήματα αποτελούνται από τρία μέρη:

- Τη βάση γνώσης
- Τη μηχανή συμπερασμού
- Τη διεπαφή με το χρήστη

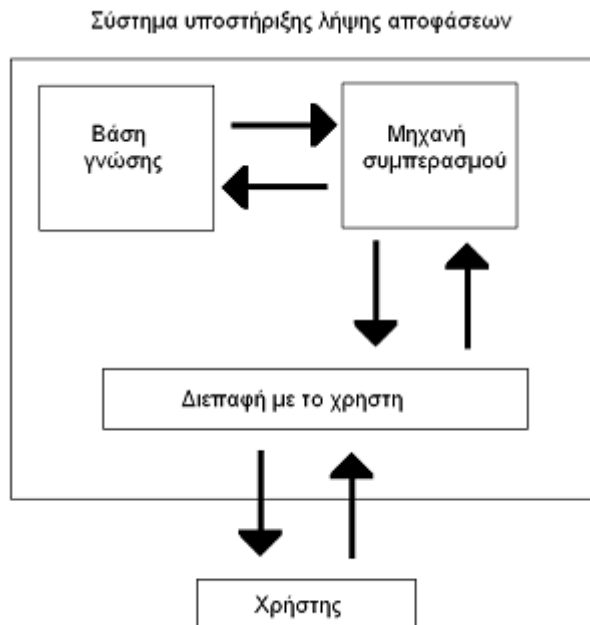
Αναλυτικότερα, η βάση γνώσης περιέχει ένα σύνολο από κανόνες του τύπου IF-THEN. Η διεπαφή δίνει τη δυνατότητα στο χρήστη να εισάγει δεδομένα στο σύστημα συνήθως μέσω ερωτήσεων του συστήματος προς το χρήστη. Η μηχανή συμπερασμού συγκρίνει τα δεδομένα του χρήστη με τα δεδομένα που υπάρχουν στη βάση γνώσης και αν κάποιος κανόνας προκύψει αληθής (**IF rule = TRUE**) τότε εκτελείται το αντίστοιχο σενάριο εντολών (**THEN {action1...action}**).

Τα συστήματα λήψης αποφάσεων μπορούν να στηρίξουν όλα τα στάδια λήψης μιας απόφασης. Εάν ο χρήστης στο πρώτο στάδιο εισάγει κάποια δεδομένα και αυτά έχουν σαν αποτέλεσμα ένα σύνολο από πιθανές απαντήσεις, τότε το σύστημα συνεχίζει να αλληλεπιδρά προσφέροντας ένα δεύτερο στάδιο εισαγωγής δεδομένων με στόχο να αποκλείσει ή να ενισχύσει πιθανά σενάρια, έτσι ώστε να δώσει ένα πιο ολοκληρωμένο και έγκυρο σύνολο απαντήσεων. Ο αριθμός των σταδίων μπορεί να διαφέρει από σύστημα σε σύστημα.

Τα οφέλη που προκύπτουν από τη χρήση αυτών των συστημάτων είναι η επίσπευση της λύσης του προβλήματος, η δυνατότητα προσομοίωσης, η εύκολη χρήση και προσαρμοστικότητα στις ανάγκες των χρηστών, η ανακάλυψη νέων προσεγγίσεων σκέψης σχετικά με το

πρόβλημα, η δημιουργία καινούργιων αποδείξεων για τη στήριξη μιας απόφασης κ.α.

Μια απλοϊκή απεικόνιση λειτουργίας ενός τέτοιου συστήματος είναι η εξής:



Ιστορία

Ήδη από τα τέλη του 1950 και αρχές του 1960 γεννήθηκε η ανάγκη για δημιουργία πληροφοριακών συστημάτων υποστήριξης λήψης αποφάσεων. Οι θεωρητικές μελέτες για την ανάπτυξή τους διενεργήθηκαν από το Ινστιτούτο τεχνολογίας του Carnegie και το Ινστιτούτο τεχνολογίας της Μασαχουσέτης. Βέβαια ο τομέας των συστημάτων υποστήριξης λήψης αποφάσεων έγινε ανεξάρτητη περιοχή έρευνας στα μέσα της δεκαετίας του 1970 όπου αναπτύχθηκαν πρακτικά τα πρώτα συστήματα υποστήριξης λήψης αποφάσεων από τους Peter Keen, Charles Stabell, Michael S. Scott Morton, Ralph Sprague και Andrew Whinston.

Εφαρμογές

Τα συστήματα υποστήριξης λήψης αποφάσεων διαθέτουν ένα ευρύ φάσμα εφαρμογής. Μερικοί από τους κλάδους στους οποίους χρησιμοποιούνται είναι η ιατρική διάγνωση, η χρηματιστηριακή αγορά, οι επιχειρήσεις και το management, η αγροτική παραγωγή, τα δίκτυα συγκοινωνιών κ.α. Γενικότερα μπορούν να χρησιμοποιηθούν σε οποιονδήποτε τομέα είναι απαραίτητη η οργάνωση.

Συστήματα λήψης ιατρικών αποφάσεων

Περιγραφή

Τα συστήματα λήψης ιατρικών αποφάσεων είναι διαδραστικά συστήματα αποφάσεων που χρησιμοποιούνται στην ιατρική διάγνωση για τη λήψη αποφάσεων. Ο ορισμός που έχει δοθεί από τον Dr. Robert Hayward του Centre for Health Evidence είναι ο εξής: Τα συστήματα λήψης ιατρικών αποφάσεων συνδέουν την παρατήρηση ασθενειών με τη γνώση σχετικά με αυτές με σκοπό να καθοδηγήσουν τις αποφάσεις των γιατρών.

Ο ρόλος αυτών των συστημάτων είναι απλά να βοηθήσουν τους γιατρούς στη λήψη αποφάσεων και όχι να αποφασίζουν αντί αυτών. Μέχρι στιγμής χρησιμοποιούνται για πολύ συγκεκριμένους σκοπούς, όπως να ερμηνεύουν τα αποτελέσματα μιας παρακολούθησης της πνευμονικής λειτουργίας, να αναλύουν ηλεκτροκαρδιογραφήματα κ.α στο μέλλον όμως αναμένεται να αυξηθεί η χρήση τους. Ο λόγος είναι οι ακριβείς και αξιόπιστες διαγνώσεις και η αποφυγή λαθών από ελλείψεις γνώσεις των γιατρών. Επίσης μειώνεται το κόστος καθώς η διάγνωση γίνεται πιο γρήγορα, οι χορηγήσεις των φαρμάκων είναι πιο αποδοτικές και εξαιρείται η ανάγκη για εξειδικευμένους γιατρούς. Βέβαια τίθεται το ερώτημα αν η κακή χρήση αυτών των συστημάτων ή κάποια λάθος λειτουργία τους, θέτει σε κίνδυνο την υγεία των ασθενών. Αυτό όμως εγείρει ερωτήματα ηθικής φύσεως τα οποία και δεν θα συζητηθούν περαιτέρω.

Χαρακτηριστικά και Μεθοδολογίες

Υπάρχουν διάφορες μεθοδολογίες που χρησιμοποιούνται στα συστήματα υποστήριξης λήψης ιατρικών αποφάσεων οι οποίες υπάγονται σε δύο κατηγορίες: τα βασισμένα στη γνώση συστήματα λήψης ιατρικών αποφάσεων και τα μη-βασισμένα στη γνώση συστήματα λήψης ιατρικών αποφάσεων. Τα πρώτα-όπως περιγράψαμε παραπάνω- αποτελούνται από τη βάση γνώσης, τη μηχανή συμπερασμού και τη διεπαφή με το χρήστη. Εισάγονται τα δεδομένα που αφορούν τον ασθενή (συμπτώματα, φάρμακα που λαμβάνει, αποτελέσματα εργαστηριακών εξετάσεων), συνδυάζονται με τους κανόνες στη μηχανή συμπερασμού και εμφανίζονται τα αποτελέσματα μέσω της διεπαφής. Τα δεύτερα, βασίζονται σε έναν κλάδο της τεχνητής νοημοσύνης, τη μηχανική μάθηση η οποία βασίζεται σε νευρωνικά δίκτυα και αλγορίθμους και επιτρέπει στους υπολογιστές να «μαθαίνουν» από προηγούμενες χρήσεις του συστήματος. Κάποιες χαρακτηριστικές μεθοδολογίες είναι:

- Bayesian δίκτυο

Το Bayesian δίκτυο είναι βασισμένο στη γνώση σύστημα υποστήριξης λήψης ιατρικών αποφάσεων και απεικονίζει γραφικά την πιθανοτική σχέση μεταξύ συμπτωμάτων και ασθενειών. Βασίζεται στη δεσμευμένη πιθανότητα, δηλαδή στην πιθανότητα ύπαρξης μιας ασθένειας δεδομένων των συμπτωμάτων.

- Νευρωνικό δίκτυο

Το νευρωνικό δίκτυο είναι μη-βασισμένο στη γνώση σύστημα υποστήριξης λήψης ιατρικών αποφάσεων και αποτελείται από ένα βεβαρημένο γράφημα το οποίο εντοπίζει τα πρότυπα (patterns) στα δεδομένα του ασθενούς και εξαγεί τη σχέση μεταξύ των συμπτωμάτων και της διάγνωσης. Λόγω του ότι το σύστημα δεν εξηγεί τον τρόπο που επεξεργάζεται τα δεδομένα, δεν χρησιμοποιείται συχνά από γιατρούς για λόγους αξιοπιστίας.

- Γενετικός αλγόριθμος

Ο γενετικός αλγόριθμος είναι κ αυτός μη-βασισμένη στη γνώση μέθοδος που αναπτύχθηκε το 1940 και βασίζεται στη θεωρία του Δαρβίνου για την επιβίωση του ισχυρότερου. Ο αλγόριθμος εκτελεί μια επαναληπτική διαδικασία η οποία κρατάει τα βέλτιστα αποτελέσματα κάθε φορά και απορρίπτει τα υπόλοιπα. Η πληροφορία που λαμβάνει το σύστημα προκύπτει κάθε φορά από τα δεδομένα του ασθενούς.

- Σύστημα βασισμένο σε κανόνες (Rule-based system)

Η λειτουργία ενός συστήματος βασισμένο σε κανόνες είναι να εξάγει συμπεράσματα από την υπάρχουσα γνώση η οποία είναι στη μορφή κανόνων. Οι κανόνες συνδυάζονται μεταξύ τους μέχρι να προκύψει ένα ικανοποιητικό αποτέλεσμα. Το πλεονέκτημα αυτής της μεθοδολογίας είναι ότι συγκεντρώνεται μεγάλη ποσότητα πληροφορίας. Το μειονέκτημα είναι ότι μπορεί να χρειαστούν πολλοί κανόνες έτσι ώστε το σύστημα να είναι αποτελεσματικό.

- Λογικές προτάσεις

Η μεθοδολογία αυτή είναι αρκετά απλοϊκή. Δεδομένης μιας μεταβλητής, ελέγχεται εάν η τιμή της βρίσκεται εντός ορίων και εκτελείται η ανάλογη πράξη. Πολλές φορές χρησιμοποιείται και πίνακας απόφασης για λόγους ευκολίας αναπαράστασης.

- Αιτιολογικό- Πιθανοτικό δίκτυο

Ένα αιτιολογικό-πιθανοτικό δίκτυο που χρησιμοποιείται στην ιατρική, αποτελείται από κόμβους και ακμές μεταξύ τους οι οποίες αναπαριστούν μια σχέση αιτίας-αποτελέσματος. Με αυτόν τον τρόπο ξεκινώντας από κόμβους-συμπτώματα καταλήγουμε σε κόμβους ασθένειες ακολουθώντας πάντα το βέλτιστο μονοπάτι.

Παραδείγματα συστημάτων υποστήριξης λήψης ιατρικών αποφάσεων

- Mycin

Το Mycin είναι ένα σύστημα βασισμένο σε κανόνες που αναπτύχθηκε το 1970 στο Stanford University και ο πρωταρχικός του

ρόλος ήταν να αναγνωρίζει τα βακτήρια που προκαλούσαν μολύνσεις του αίματος και να προτείνει αντιβιοτικά. Δεν χρησιμοποιήθηκε ποτέ για θεραπευτικούς σκοπούς, όμως είχε 69% επιτυχία στις θεραπείες που πρότεινε, ποσοστό μεγαλύτερο από αυτό που πέτυχαν ειδικοί στις λοιμώδεις ασθένειες.

Γράφτηκε σε Lisp και περιέχει πάνω από 600 κανόνες στη βάση γνώσης. Αποτελείται από απλές ερωτήσεις που απαντώνται με ένα ναι/όχι από τον θεράποντα γιατρό. Στο τέλος εμφανίζει ένα σύνολο από πιθανά βακτήρια που προκάλεσαν τα συμπτώματα, από το λιγότερο πιθανό έως το πιο πιθανό.

- Caduceus

Το Caduceus τελειοποιήθηκε στα μέσα της δεκαετίας του 1980 από τον Harry Pople του University of Pittsburgh. Σκοπός του ήταν να βελτιστοποιήσει το Mycin έτσι ώστε να μην περιορίζεται μόνο σε ασθένειες του αίματος που προκαλούνται από βακτήρια αλλά να αναγνωρίζει ένα μεγάλο εύρος ασθενειών. Μετά την ολοκλήρωσή του έφτασε να αναγνωρίζει έως και 1000 ασθένειες.

Η μηχανή συμπερασμού του Caduceus είναι παρόμοια με αυτή του Mycin αλλά με την προσθήκη υποθετικού συλλογισμού.

Θεωρείται μέχρι στιγμής το σύστημα λήψης ιατρικών αποφάσεων με την περισσότερη γνώση.

- Iliad

Το Iliad αναπτύχθηκε στο University of Utah School of Medicine's Dept. of Medical Informatics. Αρχικά σχεδιάστηκε για τη διάγνωση ασθενειών εσωτερικής παθολογίας αλλά στην πορεία συμπεριλήφθηκαν και άλλες ασθένειες. Χρησιμοποιεί τη Bayesian λογική και υπολογίζει την α αποστεριόρι πιθανότητα των διαγνώσεων λαμβάνοντας υπόψη τα δεδομένα του ασθενούς. Το Iliad καλύπτει έως και 1500 διαγνώσεις, βασιζόμενες σε χιλιάδες ευρήματα και στοιχεία

- Oncocin

Το Oncocin είναι μια εξέλιξη του Mycin και είναι σύστημα βασισμένο σε κανόνες, που είχε δημιουργηθεί για να βοηθά γιατρούς σχετικά με

καρκινοπαθείς που λάμβαναν χημειοθεραπεία συνδυάζοντας κλινικά δεδομένα με πρωτόκολλα χορήγησης της χημειοθεραπείας. Ήταν ένα από τα πρώτα συστήματα λήψης αποφάσεων που έδιναν αυτοματοποιημένες απαντήσεις

- Dxplain

Το Dxplain είναι ένα σύστημα υποστήριξης λήψης ιατρικών αποφάσεων που αναπτύχθηκε το 1980 από το Laboratory of Computer Science, Massachusetts General Hospital, Harvard Medical School. Δέχεται σαν είσοδο κλινικά ευρήματα, συμπτώματα και εργαστηριακά αποτελέσματα και εξάγει μια λίστα διαγνώσεων. Αιτιολογεί για ποιο λόγο προέκυψαν αυτά τα συμπεράσματα, ζητάει επιπλέον πληροφορίες που θα μπορούσαν να φανούν χρήσιμες και επισημάνει τυχόν ασυνήθιστα φαινόμενα που δεν συνάδουν με τη συγκεκριμένη ασθένεια. Χρησιμοποιείται και εδώ η Bayesian λογική.

Κεφάλαιο 2: Περιγραφή της εφαρμογής

Κανόνες

Για να υλοποιήσουμε ένα σύστημα λήψης ιατρικών αποφάσεων, θα χρησιμοποιήσουμε σύστημα κανόνων. Η λογική του συστήματος κανόνων είναι η εξής: ΑΝ ισχύει αυτός ο κανόνας, ΤΟΤΕ κάνε αυτή/ες την/τις ενέργεια/ες. Οι κανόνες απαρτίζονται από ένα σύνολο γεγονότων, τα οποία αν είναι αληθή, είναι και ο κανόνας αληθής. Στη συγκεκριμένη εφαρμογή που θέλουμε να υλοποιήσουμε οι κανόνες είναι το σύνολο των συμπτωμάτων και οι ενέργειες οι ασθένειες. Δηλαδή ΑΝ υπάρχουν αυτά τα συμπτώματα, ΤΟΤΕ η ασθένεια είναι αυτή. Η σύνταξη των κανόνων είναι η εξής:

- **ΑΝ {(σύμπτωμα=...) ΚΑΙ (σύμπτωμα=...) ... ΚΑΙ (σύμπτωμα=...)}**
ΤΟΤΕ {(ασθένεια=...)}

Θα χρησιμοποιήσουμε επίσης συντελεστές βεβαιότητας για κάθε κανόνα για να εκφράσουμε το βαθμό βεβαιότητας με τον οποίο δεχόμαστε το συμπέρασμα. Οι τιμές που θα δώσουμε σε κάθε κανόνα ανήκουν στο $[0,1]$ -όπου το 1 δηλώνει 100% βεβαιότητα του γεγονότος και είναι αυθαίρετες. Έτσι οι κανόνες γίνονται:

- **ΑΝ {(σύμπτωμα=...) ΚΑΙ (σύμπτωμα=...) ... ΚΑΙ (σύμπτωμα=...)}**
ΤΟΤΕ {(ασθένεια=...)(συντελεστής βεβαιότητας=...)}

Παράδειγμα

Θα παρουσιάσουμε ένα παράδειγμα για να δείξουμε τους κανόνες που εισάγονται στη βάση γνώσης. Η ασθένεια-παράδειγμα είναι η γρίπη. Θα δείξουμε ποιοι κανόνες πρέπει να ισχύουν για να βγάλουμε το συμπέρασμα ότι ένας ασθενής έχει γρίπη. (Για λόγους ευκολίας θα υποθέσουμε ότι εάν ισχύουν όλοι οι κανόνες είναι σίγουρο ότι ο ασθενής έχει γρίπη, δηλαδή ο συντελεστής βεβαιότητας είναι 1. Αυτό είναι πιθανόν να μην ισχύει στην πραγματικότητα καθώς υπάρχουν και άλλοι παράγοντες που επηρεάζουν τη διάγνωση, όμως στην προκειμένη δεν μας αφορά κάτι τέτοιο.) Έτσι έχουμε:

AN

**{ (σύμπτωμα=πυρετός) ΚΑΙ (σύμπτωμα=βήχας) ΚΑΙ
(σύμπτωμα=πονοκέφαλος) ΚΑΙ (σύμπτωμα=ρίγος) ΚΑΙ
(σύμπτωμα=κόπωση) }**

TOTE

{ (ασθένεια=γρίπη) (συντελεστής βεβαιότητας=1) }

Αυτός είναι ο γενικός κανόνας που μας εξασφαλίζει ότι εάν ο ασθενής έχει τα παραπάνω συμπτώματα, τότε σίγουρα έχει γρίπη. Τι γίνεται όμως αν ασθενής παρουσιάζει ένα ή τέσσερα από τα συμπτώματα; Είναι πιθανόν να έχει γρίπη, και αν ναι, με τι συντελεστή βεβαιότητας; Για να έχουμε απάντηση για όλες τις δυνατές περιπτώσεις, πρέπει να εισάγουμε και άλλους κανόνες. Παρακάτω είναι όλοι οι κανόνες που έχουν σαν συμπέρασμα την γρίπη:

- **AN**

{ (σύμπτωμα=πυρετός)}

TOTE

{ (ασθένεια=γρίπη) (συντελεστής βεβαιότητας=0.2) }

- **AN**

{(σύμπτωμα=βήχας)}

TOTE

{(ασθένεια=γρίπη)(συντελεστής βεβαιότητας=0.1)}

- **AN**

{(σύμπτωμα=πονοκέφαλος)}

TOTE

{(ασθένεια=γρίπη)(συντελεστής βεβαιότητας=0.1)}

- **AN**

{ (σύμπτωμα=ρίγος) }

TOTE

{ (ασθένεια=γρίπη) (συντελεστής βεβαιότητας=0.3) }

- AN

{ (σύμπτωμα=κόπωση) }

TOTE

{ (ασθένεια=γρίπη) (συντελεστής βεβαιότητας=0.1) }

- AN

{ (σύμπτωμα=πυρετός) ΚΑΙ (σύμπτωμα=βήχας) }

TOTE

{ (ασθένεια=γρίπη) (συντελεστής βεβαιότητας=0.3) }

- AN

{ (σύμπτωμα=πυρετός) ΚΑΙ (σύμπτωμα=ρίγος) }

TOTE

{ (ασθένεια=γρίπη) (συντελεστής βεβαιότητας=0.5) }

- AN

{ (σύμπτωμα=πυρετός) ΚΑΙ (σύμπτωμα=πονοκέφαλος) }

TOTE

{ (ασθένεια=γρίπη) (συντελεστής βεβαιότητας=0.5) }

- AN

{ (σύμπτωμα=βήχας) ΚΑΙ (σύμπτωμα=πονοκέφαλος) }

TOTE

{ (ασθένεια=γρίπη) (συντελεστής βεβαιότητας=0.4) }

- AN

{ (σύμπτωμα=κόπωση) ΚΑΙ (σύμπτωμα=πονοκέφαλος) }

TOTE

{ (ασθένεια=γρίπη) (συντελεστής βεβαιότητας=0.4) }

- AN

{ (σύμπτωμα=βήχας) ΚΑΙ (σύμπτωμα=ρίγος) }

TOTE

{ (ασθένεια=γρίπη) (συντελεστής βεβαιότητας=0.4) }

- AN

{ (σύμπτωμα=κόπωση) ΚΑΙ (σύμπτωμα=ρίγος) }

TOTE

{ (ασθένεια=γρίπη) (συντελεστής βεβαιότητας=0.4) }

- AN

{ (σύμπτωμα=πονοκέφαλος) ΚΑΙ (σύμπτωμα=ρίγος) }

TOTE

{ (ασθένεια=γρίπη) (συντελεστής βεβαιότητας=0.6) }

- AN

{ (σύμπτωμα=κόπωση) ΚΑΙ (σύμπτωμα=βήχας) }

TOTE

{ (ασθένεια=γρίπη) (συντελεστής βεβαιότητας=0.2) }

- AN

{ (σύμπτωμα=πυρετός) ΚΑΙ (σύμπτωμα=κόπωση) }

TOTE

{ (ασθένεια=γρίπη) (συντελεστής βεβαιότητας=0.3) }

- AN

{ (σύμπτωμα=πυρετός) ΚΑΙ (σύμπτωμα=κόπωση) ΚΑΙ
(σύμπτωμα=βήχας) }

TOTE

{ (ασθένεια=γρίπη) (συντελεστής βεβαιότητας=0.4) }

- AN

{ (σύμπτωμα=πυρετός) ΚΑΙ (σύμπτωμα=πονοκέφαλος) ΚΑΙ
(σύμπτωμα=βήχας) }

TOTE

{ (ασθένεια=γρίπη) (συντελεστής βεβαιότητας=0.6) }

- AN

{ (σύμπτωμα=πυρετός) ΚΑΙ (σύμπτωμα=κόπωση) ΚΑΙ
(σύμπτωμα=πονοκέφαλος) }

TOTE

{ (ασθένεια=γρίπη) (συντελεστής βεβαιότητας=0.6) }

- AN

{ (σύμπτωμα=πονοκέφαλος) ΚΑΙ (σύμπτωμα=κόπωση) ΚΑΙ
(σύμπτωμα=βήχας) }

TOTE

{ (ασθένεια=γρίπη) (συντελεστής βεβαιότητας=0.5) }

- AN

{ (σύμπτωμα=πυρετός) ΚΑΙ (σύμπτωμα=κόπωση) ΚΑΙ
(σύμπτωμα=ρίγος) }

TOTE

{ (ασθένεια=γρίπη) (συντελεστής βεβαιότητας=0.6) }

- AN

{ (σύμπτωμα=πυρετός) ΚΑΙ (σύμπτωμα=βήχας) ΚΑΙ (σύμπτωμα=ρίγος)
}

TOTE

{ (ασθένεια=γρίπη) (συντελεστής βεβαιότητας=0.6) }

- AN

{ (σύμπτωμα=κόπωση) ΚΑΙ (σύμπτωμα=βήχας) ΚΑΙ (σύμπτωμα=ρίγος)
}

TOTE

{ (ασθένεια=γρίπη) (συντελεστής βεβαιότητας=0.5) }

- AN

{ (σύμπτωμα=πυρετός) ΚΑΙ (σύμπτωμα=πονοκέφαλος) ΚΑΙ
(σύμπτωμα=ρίγος) }

TOTE

{ (ασθένεια=γρίπη) (συντελεστής βεβαιότητας=0.8) }

- AN

{ (σύμπτωμα=βήχας) ΚΑΙ (σύμπτωμα=βήχας) ΚΑΙ (σύμπτωμα=ρίγος) }

TOTE

{ (ασθένεια=γρίπη) (συντελεστής βεβαιότητας=0.7) }

- AN

{ (σύμπτωμα=κόπωση) ΚΑΙ (σύμπτωμα=πονοκέφαλος) ΚΑΙ
(σύμπτωμα=ρίγος) }

TOTE

{ (ασθένεια=γρίπη) (συντελεστής βεβαιότητας=0.7) }

- AN

{ (σύμπτωμα=πυρετός) ΚΑΙ (σύμπτωμα=βήχας) ΚΑΙ
(σύμπτωμα=κόπωση) ΚΑΙ (σύμπτωμα=ρίγος) }

TOTE

{ (ασθένεια=γρίπη) (συντελεστής βεβαιότητας=0.7) }

- AN

{ (σύμπτωμα=πυρετός) ΚΑΙ (σύμπτωμα=βήχας) ΚΑΙ
(σύμπτωμα=κόπωση) ΚΑΙ (σύμπτωμα=πονοκέφαλος) }

TOTE

{ (ασθένεια=γρίπη) (συντελεστής βεβαιότητας=0.7) }

- AN

{ (σύμπτωμα=πυρετός) ΚΑΙ (σύμπτωμα=βήχας) ΚΑΙ
(σύμπτωμα=πονοκέφαλος) ΚΑΙ (σύμπτωμα=ρίγος) }

TOTE

{ (ασθένεια=γρίπη) (συντελεστής βεβαιότητας=0.9) }

- AN

{ (σύμπτωμα=πονοκέφαλος) ΚΑΙ (σύμπτωμα=βήχας) ΚΑΙ
(σύμπτωμα=κόπωση) ΚΑΙ (σύμπτωμα=ρίγος) }

TOTE

{ (ασθένεια=γρίπη) (συντελεστής βεβαιότητας=0.8) }

- AN

{ (σύμπτωμα=πυρετός) ΚΑΙ (σύμπτωμα=πονοκέφαλος) ΚΑΙ
(σύμπτωμα=κόπωση) ΚΑΙ (σύμπτωμα=ρίγος) }

TOTE

{ (ασθένεια=γρίπη) (συντελεστής βεβαιότητας=0.9) }

- AN

{ (σύμπτωμα=πυρετός) ΚΑΙ (σύμπτωμα=βήχας) ΚΑΙ
(σύμπτωμα=κόπωση) ΚΑΙ (σύμπτωμα=ρίγος) ΚΑΙ
(σύμπτωμα=πονοκέφαλος)}

TOTE

{ (ασθένεια=γρίπη) (συντελεστής βεβαιότητας=1) }

Κεφάλαιο 3: Υλοποίηση του προγράμματος

Το πρόγραμμα CLIPS

Το CLIPS είναι ένα εργαλείο για τη δημιουργία expert systems. Αναπτύχθηκε το 1985 στο NASA-Johnson Space Center και έχει γραφτεί σε γλώσσα C, ενώ η διεπαφή του θυμίζει αρκετά τη γλώσσα LISP. Αποτελείται από ένα παράθυρο διαλόγου με το οποίο αλληλεπιδρά ο χρήστης. Τα αρχεία που δημιουργούνται έχουν προέκταση .clp.

Κανόνες και γεγονότα

Η λειτουργία του CLIPS περιλαμβάνει κανόνες οι οποίοι περιέχουν γεγονότα. Εάν τα γεγονότα είναι αληθή, ο κανόνας ισχύει και εκτελείται μια ακολουθία εντολών. Η σύνταξη των κανόνων έχει ως εξής:

(defrule name

(fact)

=>

(actions))

Ένα παράδειγμα:

(defrule ost1.1

(symptom loss_of_height)

=>

(assert (disease osteoporosis)))

Η έκφραση **defrule** ορίζει τον κανόνα. **Ost1** είναι το όνομα του κανόνα. Το γεγονός είναι το **(symptom loss_of_height)** το οποίο εάν είναι αληθές, δηλαδή εάν υπάρχει στη βάση γνώσης τότε ενεργοποιείται ο κανόνας και εκτελείται η ακολουθία εντολών μετά το =>. Για να υπάρχει ένα γεγονός στη βάση γνώσης πρέπει πρωτίτερα να έχει εισαχθεί. Ο

τρόπος για να εισαχθεί ένα γεγονός γίνεται με την εντολή **assert**, όπως στο παράδειγμά μας για να είναι αληθές το **(symptom loss_of_height)** θα πρέπει να έχει δοθεί πρωτύτερα η εντολή **(assert (symptom loss_of_height))**. Αντίστοιχα εισάγεται στη βάση γνώσης το γεγονός **(disease osteoporosis)** σαν συνέπεια του κανόνα **ost1.1**.

Φυσικά σε έναν κανόνα μπορούν να υπάρχουν πολλά γεγονότα που πρέπει να ισχύουν ταυτόχρονα, όπως και πλήθος εντολών προς εκτέλεση. Παράδειγμα:

(defrule s2.3

(symptom fever)

(symptom tiredness)

=>

(assert (disease pneumonia))

(assert (disease flu)))

Για να είναι αληθής ο κανόνας πρέπει να ισχύουν και τα δύο αυτά γεγονότα.

Προτεραιότητα κανόνων

Όταν ο κώδικάς μας περιλαμβάνει πολλούς κανόνες οι οποίοι περιέχουν κάποια κοινά γεγονότα τα οποία είναι αληθή, τότε ποιος κανόνας θα ενεργοποιηθεί; Θα ενεργοποιηθεί οποιοσδήποτε από αυτούς αλλά το ερώτημα είναι ποιος θέλουμε να ενεργοποιηθεί; Παράδειγμα:

(defrule s3.1

(symptom fever)

(symptom cough)

(symptom tiredness)

=>

(assert (symptom pneumonia)))

Και

(defrule s2.3

(symptom fever)

(symptom tiredness)

=>

(assert (disease flu))

(assert (symptom pneumonia)))

Εάν υποθέσουμε ότι όλα τα γεγονότα και των δύο κανόνων είναι αληθή, τότε θα ενεργοποιηθούν και οι δύο κανόνες και συνεπώς θα εισαχθούν στη βάση γνώσης όλα τα επακόλουθα γεγονότα. Έστω όμως ότι θέλουμε να ενεργοποιήσουμε μόνο τον κανόνα s3.1 μιας και περιέχει το επιπλέον γεγονός **(symptom cough)**, κάτι το οποίο αποκλείει το γεγονός **(disease flu)** το οποίο δεν θέλουμε να συμπεριλάβουμε στη βάση γνώσης. Με τη συγκεκριμένη σύνταξη των κανόνων θα ενεργοποιηθούν και οι δύο. Για να αποτρέψουμε κάτι τέτοιο θα θέσουμε προτεραιότητα στους κανόνες με την εντολή **(declare (salience X))** όπου X ένας αριθμός που καθορίζει την προτεραιότητα ενός κανόνα. Όσο μεγαλύτερος ο αριθμός, τόσο μεγαλύτερη και η προτεραιότητα του κανόνα. Επιστρέφοντας στο παράδειγμά μας, ο κανόνας s3.1 γίνεται:

(defrule s3.1

(declare (salience 30))

(symptom fever)

(symptom cough)

(symptom tiredness)

=>

(assert (symptom pneumonia)))

Και ο κανόνας s2.3:

(defrule s2.3

(declare (salience 20))

(symptom fever)

(symptom tiredness)

=>

(assert (disease flu))

(assert (symptom pneumonia)))

Έτσι εξασφαλίζουμε ότι θα ενεργοποιηθεί μόνο ο κανόνας s3.1.

Αλληλεπίδραση με το χρήστη

- Εισαγωγή από το πληκτρολόγιο

Για να μπορεί το σύστημά μας να λειτουργεί σωστά, θα πρέπει κάποιος να εισάγει τα γεγονότα που ενεργοποιούν τους κανόνες. Αυτό γίνεται είτε κατά τη συγγραφή του προγράμματος με την εντολή **(assert (fact))**, όπου τα γεγονότα που εισάγουμε μπαίνουν εξ αρχής στη βάση γνώσης, είτε σαν αποτέλεσμα κανόνα (**=> (assert (fact))**), είτε εισάγονται από το χρήστη του προγράμματος κατά τη διάρκεια της εκτέλεσής του. Αυτό γίνεται με τις εξής εντολές:

(bind ?sym(read))

(assert (symptom ?sym)))

Η εντολή **read** σημαίνει ότι μπορούμε να δώσουμε δεδομένα από το πληκτρολόγιο τα οποία μέσω της εντολής **bind** αποθηκεύονται στη μεταβλητή **sym**. Προσοχή: το όνομα της μεταβλητής είναι **sym** και όχι **?sym**. Το **?** μπαίνει μπροστά από το όνομα κάθε μεταβλητής. Έπειτα το περιεχόμενο της μεταβλητής **sym** εισάγεται σαν γεγονός στη βάση γνώσης με την εντολή **assert**.

- Εμφάνιση στην οθόνη

Για να κάνουμε το πρόγραμμά μας πιο λειτουργικό και να μπορέσουμε να δώσουμε στο χρήστη να καταλάβει πότε πρέπει να εισάγει τα δεδομένα που επιθυμεί, πρέπει το πρόγραμμά μας να εμφανίζει τα κατάλληλα μηνύματα στην οθόνη. Αυτό επιτυγχάνεται με την εντολή :

(printout t "text" crlf)

όπου **printout** είναι η εντολή εμφάνισης, το **t** δηλώνει το **terminal**, δηλαδή την εμφάνιση του μηνύματος στην οθόνη, **text** είναι το κείμενο που επιθυμούμε να εισάγουμε και το **crlf** υποδηλώνει την αλλαγή γραμμής . Ας δούμε ένα απλό παράδειγμα που να περιλαμβάνει όλες τις εντολές που αναφέραμε πιο πάνω:

(defrule pn2.1

(declare (salience 20))

(symptom tiredness)

(symptom breathlessness)

=>

(printout t "You have pneumonia with certainty factor 0.5" crlf)

(printout t "Enter "end" if you don't have other symptoms" crlf)

(printout t "You might also have one of the following symptoms:" crlf)

(printout t "cough, fever" crlf)

(printout t "enter which one" crlf)

(bind ?sym(read))

(assert (pn.pos 0.5))

(assert (symptom ?sym))

(assert (disease pneumonia)))

Ο συγκεκριμένος κανόνας αποτελείται από το «υποθετικό» κομμάτι, δηλαδή από τα γεγονότα **(symptom tiredness)** και **(symptom**

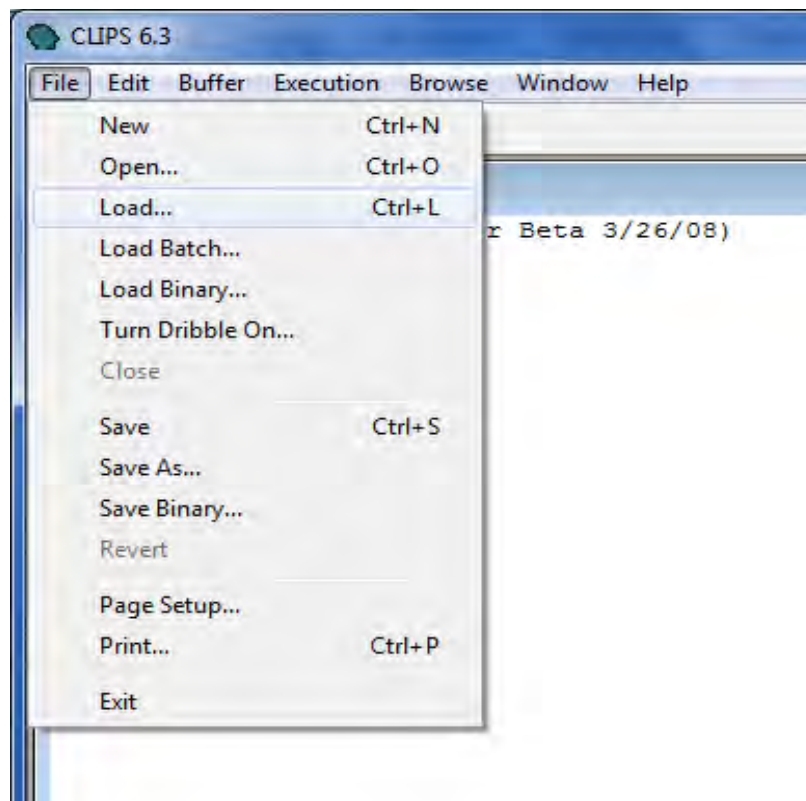
breathlessness) τα οποία αν ισχύουν ακολουθεί το μπλοκ των εντολών (**printout t "text" crlf**), οι εντολές εισαγωγής από το πληκτρολόγιο και η εισαγωγή επιπλέον γεγονότων. Αυτός ο κανόνας ενημερώνει το χρήστη πως αν έχει εισάγει τα συμπτώματα **firedness** και **breathlessness**, τότε έχει πνευμονία με συντελεστή βεβαιότητας 0.5. Επίσης δίνει τη δυνατότητα στο χρήστη να εισάγει κάποιο από τα συμπτώματα **cough** και **fever** εάν αυτά υπάρχουν.

Ενεργοποίηση των κανόνων

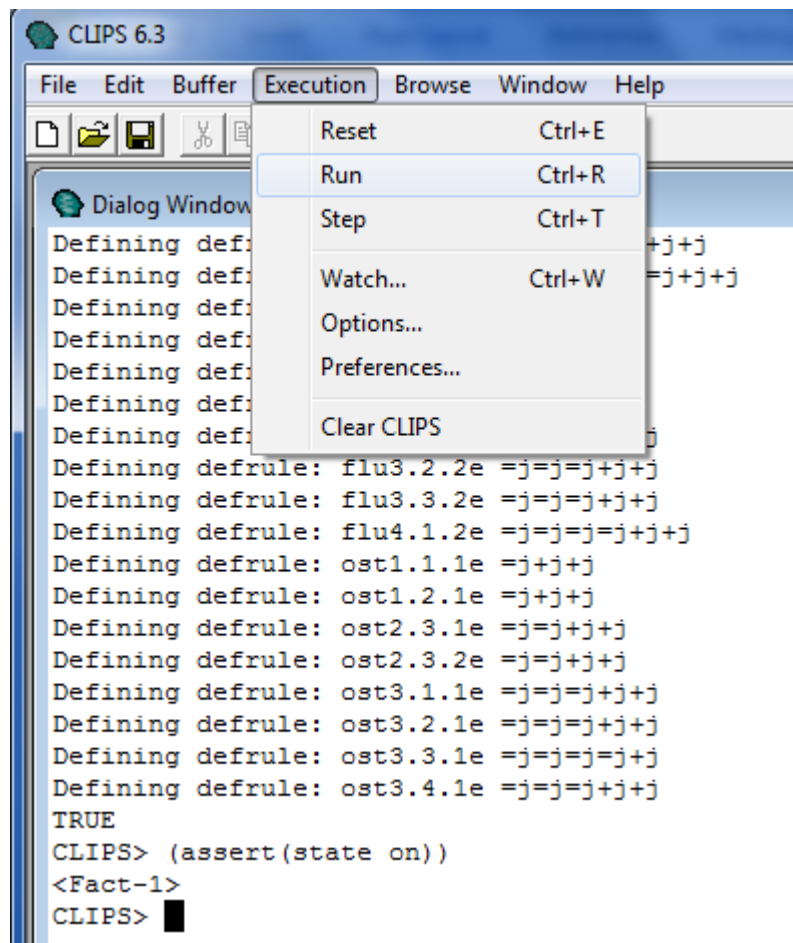
Για να εκτελεστεί το πρόγραμμα πρέπει πρώτα να φορτώσουμε όλα τα αρχεία .clp τα οποία περιέχουν τον κώδικα, με την εντολή **file->load**. Αφού έχουμε μεταβεί στο παράθυρο διαλόγου, πρέπει με κάποιο τρόπο να ενεργοποιήσουμε το πρόγραμμα και να ξεκινήσουμε την εκτέλεσή του. Αρχικά πρέπει να εισάγουμε το γεγονός **state on**. Αυτό γίνεται με τη γνωστή εντολή (**assert (state on)**) την οποία πληκτρολογούμε στη γραμμή εντολών του CLIPS. Αφού έχει αποθηκευτεί το γεγονός στη βάση γνώσης, από το μενού **execution** δίνουμε την εντολή **run**. Ο κώδικάς μας είναι γραμμένος έτσι ώστε σε κάθε κανόνα να εισάγεται από το πληκτρολόγιο ένα καινούργιο γεγονός, το οποίο με την προσθήκη του ενεργοποιεί κάποιον άλλο κανόνα. Έτσι με την προσθήκη του **state on** και την έναρξη εκτέλεσης (**run**) ενεργοποιήθηκε το πρόγραμμα.

Ένα στιγμιότυπο εκτέλεσης του προγράμματος

Ας δούμε παρακάτω πως εκτελείται το πρόγραμμα και τον κώδικα πίσω από την εκτέλεση. Πρώτα φορτώσαμε τα αρχεία .clp με την εντολή **load**



Αφού φορτώσουμε όλα τα αρχεία με τον κώδικα των κανόνων που έχουμε γράψει, γράφουμε στο παράθυρο διαλόγου (`assert(state on)`) και πατάμε `enter` ωστό να εισαχθεί το γεγονός `state on` στη βάση γνώσης. Έπειτα από το μενού **execution** επιλέγουμε **run**, όπως φαίνεται παρακάτω:



Σαν αποτέλεσμα παίρνουμε αυτό:

```
CLIPS> (assert(state on))
<Fact-1>
CLIPS> (run)
FIRE 1 start: f-1
Insert one of the following symptoms:
bone_fracture
back_pain
curvature
loss_of_height
fever
cough
tiredness
breathlessness
chills
headache
sneeze
sore_throat
Or end if you end the session
```

Βλέπουμε ότι το πρώτο γεγονός εισήχθη στη βάση γνώσης ως <Fact – 1> και στη συνέχεια ενεργοποιήθηκε ο πρώτος κανόνας start μέσω του γεγονότος f-1. Ο κανόνας που ενεργοποιείται φαίνεται στη γραμμή FIRE

ακολουθούμενος από τα γεγονότα που τον ενεργοποίησαν. Ο κανόνας που ενεργοποιήθηκε είναι ο εξής:

```
(defrule start
```

```
(state on) //εάν περιέχεται το γεγονός αυτό στη βάση γνώσης
```

```
=> //εκτέλεσε τα ακόλουθα
```

```
(printout t "Insert one of the following symptoms:" crlf)
```

```
(printout t
```

```
"bone_fracture
```

```
back_pain
```

```
curvature
```

```
loss_of_height
```

```
fever
```

```
cough
```

```
tiredness
```

```
breathlessness
```

```
chills
```

```
headache
```

```
sneeze
```

```
sore_throat" crlf)
```

```
(printout t "Or "end" if you end the session" crlf)
```

```
(bind ?sym(read))
```

```
(assert (symptom ?sym)))
```

Τώρα επιλέγουμε να δώσουμε κάποιο σύμπτωμα από τη λίστα. Πληκτρολογούμε fever και πατάμε enter.

```

fever
FIRE 2 s1.1: f-2
You have pneumonia with certainty factor 0.1, or flu with certainty factor 0.2 or common cold with certainty factor 0.1
Enter end if you don't have other symptoms
You might also have one of the following symptoms:
cough
tiredness
breathlessness
chills
headache
sneeze
sore_throat
enter which one

```

Εφ'όσον εισήχθη και το **(symptom fever)** στη βάση γνώσης σαν f-2 βλέπουμε ότι ενεργοποίησε τον κανόνα **s1.1**. Ο κανόνας s1.1 είναι:

```
(defrule s1.1
```

```
(declare (salience 10))
```

```
(symptom fever)
```

```
=>
```

```
(printout t "You have pneumonia with certainty factor 0.1, or flu with
certainty factor 0.2 or common cold with certainty factor 0.1" crlf)
```

```
(printout t "Enter "end" if you don't have other symptoms" crlf)
```

```
(printout t "You might also have one of the following symptoms:" crlf)
```

```
(printout t
```

```
"cough
```

```
tiredness
```

```
breathlessness
```

```
chills
```

```
headache
```

```
sneeze
```

```
sore_throat" crlf)
```

```
(printout t "enter which one" crlf)
```

```
(bind ?sym(read))

(assert (cf_pn 0.1)) //εισαγωγή των f-3

(assert (cf_flu 0.2)) //f-4

(assert (cf_cc 0.1)) //f-5

(assert (symptom ?sym))) //f-6 στη βάση γνώσης
```

Τώρα οι πιθανές ασθένειες είναι 3 και πρέπει να συνεχίσουμε να εισάγουμε συμπτώματα από τη λίστα. Εισάγουμε chills.

```
chills
FIRE      3 flu2.1: f-2,f-6
You have flu with certainty factor 0.5 or common cold with certainty factor 0.2
Enter end if you don't have other symptoms
You might also have one of the following symptoms:
cough
tiredness
headache
sneeze
sore_throat
enter which one
█
```

Το γεγονός f-6 (**symptom chills**) απέκλεισε το ενδεχόμενο της πνευμονίας και μαζί με το f-2 (**symptom fever**) ενεργοποίησαν τον κανόνα **flu2.1** ο οποίος είναι:

```
(defrule flu2.1

(declare (salience 20)) //μεγαλύτερη προτεραιότητα από τον

(symptom fever)          // προηγούμενο κανόνα

(symptom chills)

=>

(printout t "You have flu with certainty factor 0.5 or common cold with
certainty factor 0.2" crlf)

(printout t "Enter "end" if you don't have other symptoms" crlf)

(printout t "You might also have one of the following symptoms:" crlf)

(printout t

"cough
```

tiredness

headache

sneeze

sore_throat" crlf)

(printout t "enter which one" crlf)

(bind ?sym(read))

(assert (cf_flu 0.5)) //εισαγωγή f-7

(assert (cf_cc 0.2)) //f-8

(assert (symptom ?sym))) //f-9 στη βάση γνώσης

Θα συνεχίσουμε και με άλλα συμπτώματα, ένα εκ των οποίων το sore_throat.

```
sore_throat
FIRE      4 cc3.3.2: f-2,f-6,f-9
You have common cold with certainty factor 0.45
Enter end if you don't have other symptoms
You might also have one of the following symptoms:
cough
tiredness
headache
sneeze
enter which one
```

Εισάγοντας το γεγονός f-9 (**symptom sore_throat**) αποκλείεται το ενδεχόμενο της γρίπης και ενεργοποιείται ο κανόνας **cc3.3.2**.

(defrule cc3.3.2

(declare (salience 30))

(symptom fever)

(symptom chills)

(symptom sore_throat)

=>

(printout t "You have common cold with certainty factor 0.45" crlf)

(printout t "Enter "end" if you don't have other symptoms" crlf)

(printout t "You might also have one of the following symptoms:" crlf)

```

(printout t
"cough
tiredness
headache
sneeze" crlf)

(printout t "enter which one" crlf)

(bind ?sym(read))

(assert (cf_cc 0.45)) //εισαγωγή f-10

(assert (symptom ?sym))) //f-11 στη βάση γνώσης

```

Συνεχίζουμε να εισάγουμε και άλλα συμπτώματα για να δούμε πως εκτελείται το πρόγραμμα. Εισάγουμε sneeze.

```

sneeze
FIRE      5 c4.13: f-2,f-6,f-11,f-9
You have common cold with certainty factor 0.7
Enter end if you don't have other symptoms
You might also have one of the following symptoms:
cough
tiredness
headache
enter which one
■

```

Μετά την εισαγωγή του f-11 ενεργοποιήθηκε ο κανόνας **c4.13**. Πλέον είναι πιο πιθανή η ύπαρξη του κρυολογήματος καθώς ο συντελεστής βεβαιότητας αυξήθηκε σε 0.7.

```

(defrule c4.13

(declare (salience 40))

(symptom fever)

(symptom chills)

(symptom sneeze)

(symptom sore_throat)

=>

(printout t "You have common cold with certainty factor 0.7" crlf)

```

```

(printout t "Enter \"end\" if you don't have other symptoms" crlf)
(printout t "You might also have one of the following symptoms:" crlf)
(printout t
"cough
tiredness
headache" crlf)
(printout t "enter which one" crlf)
(bind ?sym(read))
(assert (cf_cc 0.7))
(assert (symptom ?sym)))

```

Θα εισάγουμε και τα υπόλοιπα συμπτώματα από τη λίστα για λόγους παρουσίασης.

```

cough
FIRE      6 cc5.7: f-2,f-13,f-6,f-11,f-9
You have common cold with certainty factor 0.8
Enter end if you don't have other symptoms
You might also have one of the following symptoms:
tiredness
headache
enter which one
█

```

Ο κανόνας **cc5.7**:

```

(defrule cc5.7
(declare (salience 50))
(symptom fever)
(symptom cough)
(symptom chills)
(symptom sneeze)
(symptom sore_throat)

```

=>


```

(printout t "You have common cold with certainty factor 0.8" crlf)
(printout t "Enter \"end\" if you don't have other symptoms" crlf)
(printout t "You might also have one of the following symptoms:" crlf)
(printout t
"tiredness
headache" crlf)
(printout t "enter which one" crlf)
(bind ?sym(read))
(assert (cf_cc 0.8))
(assert (symptom ?sym)))

```

```

tiredness
FIRE      7 cc6.2: f-13,f-15,f-2,f-6,f-11,f-9
You have common cold with certainty factor 0.9
Enter end if you don't have other symptoms
You might also have one of the following symptoms:
headache
enter which one

```

Ο κανόνας **cc6.2**:

```

(defrule cc6.2
(declare (salience 60))
(symptom cough)
(symptom tiredness)
(symptom fever)
(symptom chills)
(symptom sneeze)
(symptom sore_throat)

```

=>

```

(printout t "You have common cold with certainty factor 0.9" crlf)

```

```

(printout t "Enter \"end\" if you don't have other symptoms" crlf)
(printout t "You might also have one of the following symptoms:" crlf)
(printout t "headache" crlf)
(printout t "enter which one" crlf)
(bind ?sym(read))
(assert (cf_cc 0.9))
(assert (symptom ?sym)))

```

```

headache
FIRE      8 cc7: f-17,f-2,f-15,f-13,f-6,f-11,f-9
You have common cold with certainty factor 1
Enter end
■

```

Ο κανόνας **cc7**:

```

(defrule cc7
(declare (salience 70))
(symptom headache)
(symptom fever)
(symptom tiredness)
(symptom cough)
(symptom chills)
(symptom sneeze)
(symptom sore_throat)

```

=>

```

(printout t "You have common cold with certainty factor 1" crlf)
(printout t "Enter \"end\" " crlf)
(bind ?sym(read))
(assert (cf_cc 1))

```

Σε αυτό το σημείο έχουμε σίγουρα επιβεβαιώσει ότι η ασθένειά μας είναι το κρυολόγημα, μιας και ο συντελεστής βεβαιότητας είναι 1. Για να τερματίσουμε το πρόγραμμα πληκτρολογούμε end.

```
end
FIRE      9 cc7e: f-17,f-2,f-15,f-13,f-6,f-11,f-9,f-19
You have common cold with certainty factor 1
End of session
CLIPS>
```

Ο κανόνας **cc7e**:

```
(defrule cc7e
```

```
(declare (salience 71))
```

```
(symptom headache)
```

```
(symptom fever)
```

```
(symptom tiredness)
```

```
(symptom cough)
```

```
(symptom chills)
```

```
(symptom sneeze)
```

```
(symptom sore_throat)
```

```
(symptom end)
```

```
=>
```

```
(printout t "You have common cold with certainty factor 1" crlf)
```

```
(printout t "End of session" crlf)
```

```
(assert(disease common cold))
```

```
(reset)) //με αυτή την εντολή τερματίζουμε το πρόγραμμα και  
σβήνουμε οποιοδήποτε γεγονός έχουμε εισάγει στη βάση γνώσης.
```

Αυτό ήταν ένα παράδειγμα εκτέλεσης του προγράμματος. Με εισαγωγή διαφορετικών γεγονότων θα παίρναμε διαφορετικό αποτέλεσμα. Επίσης

είναι δυνατός ο τερματισμός του προγράμματος σε οποιαδήποτε στιγμή, ανεξάρτητα από των αριθμό των συμπτωμάτων που εισήγαμε και χωρίς να έχουμε καταλήξει σε ένα μόνο συμπέρασμα και είναι πιθανό να τερματίσει το πρόγραμμα χωρίς να είναι ο συντελεστής βεβαιότητας ίσος με 1.

Ας δούμε τώρα ένα ακόμα στιγμιότυπο εκτέλεσης του προγράμματος για να δείξουμε πως μπορεί να τερματίσει το πρόγραμμα χωρίς να έχει καταλήξει σε ένα μόνο συμπέρασμα και με συντελεστή βεβαιότητας 1.

Ξεκινώντας εισάγοντας το γεγονός (**state on**) ενεργοποιούμε την εκτέλεση όπως και πορηγουμένως και παίρνουμε πάλι:

```
CLIPS> (assert(state on))
<Fact-1>
CLIPS> (run)
FIRE    1 start: f-1
Insert one of the following symptoms:
bone_fracture
back_pain
curvature
loss_of_height
fever
cough
tiredness
breathlessness
chills
headache
sneeze
sore_throat
Or end if you end the session
█
```

(Ο κώδικας είναι ίδιος με παραπάνω.)

Εισάγουμε tiredness.

```
tiredness
FIRE    2 s1.3: f-2
You have pneumonia with certainty factor 0.2, or flu with certainty factor 0.1 or common cold with certainty factor 0.1
Enter end if you don't have other symptoms
You might also have one of the following symptoms:
cough
fever
breathlessness
chills
headache
sneeze
sore_throat
enter which one
█
```

Ο κανόνας **s1.3** είναι:

```
(defrule s1.3
```

```
(declare (salience 10))
```

```
(symptom tiredness)
```

```
=>
```

```
(printout t "You have pneumonia with certainty factor 0.2, or flu with  
certainty factor 0.1 or common cold with certainty factor 0.1" crlf)
```

```
(printout t "Enter \"end\" if you don't have other symptoms" crlf)
```

```
(printout t "You might also have one of the following symptoms:" crlf)
```

```
(printout t
```

```
"cough
```

```
fever
```

```
breathlessness
```

```
chills
```

```
headache
```

```
sneeze
```

```
sore_throat" crlf)
```

```
(printout t "enter which one" crlf)
```

```
(bind ?sym(read))
```

```
(assert (cf_pn 0.2))
```

```
(assert (cf_flu 0.1))
```

```
(assert (cf_cc 0.1))
```

```
(assert (symptom ?sym)))
```

Εισάγουμε τώρα cough:

```
cough
FIRE 3 s2.2: f-2,f-6
You have pneumonia with certainty factor 0.4, or flu with certainty factor 0.2 or common cold with certainty factor 0.2
Enter end if you don't have other symptoms
You might also have one of the following symptoms:
fever
breathlessness
chills
headache
sneeze
sore_throat
enter which one
```

Ο κανόνας **s2.2** είναι:

```
(defrule s2.2
```

```
(declare (salience 20))
```

```
(symptom tiredness)
```

```
(symptom cough)
```

```
=>
```

```
(printout t "You have pneumonia with certainty factor 0.4, or flu with
certainty factor 0.2 or common cold with certainty factor 0.2" crlf)
```

```
(printout t "Enter "end" if you don't have other symptoms" crlf)
```

```
(printout t "You might also have one of the following symptoms:" crlf)
```

```
(printout t
```

```
"fever
```

```
breathlessness
```

```
chills
```

```
headache
```

```
sneeze
```

```
sore_throat" crlf)
```

```
(printout t "enter which one" crlf)
```

```

(bind ?sym(read))
(assert (cf_pn 0.4))
(assert (cf_flu 0.2))
(assert (cf_cc 0.2))
(assert (symptom ?sym)))

```

Εισάγουμε fever:

```

fever
FIRE 4 s3.1: f-10,f-6,f-2
You have pneumonia with certainty factor 0.5, or flu with certainty factor 0.4 or common cold with certainty factor 0.3
Enter end if you don't have other symptoms
You might also have one of the following symptoms:
breathlessness
chills
headache
sneeze
sore_throat
enter which one
█

```

Ο κανόνας **s3.1** είναι:

```

(defrule s3.1
(declare (salience 30))
(symptom fever)
(symptom cough)
(symptom tiredness)
=>
(printout t "You have pneumonia with certainty factor 0.5, or flu with
certainty factor 0.4 or common cold with certainty factor 0.3" crlf)
(printout t "Enter \"end\" if you don't have other symptoms" crlf)
(printout t "You might also have one of the following symptoms:" crlf)
(printout t
"breathlessness

```

```

chills
headache
sneeze
sore_throat" crlf)

(printout t "enter which one" crlf)

(bind ?sym(read))

(assert (cf_pn 0.5))

(assert (cf_flu 0.4))

(assert (cf_cc 0.3))

(assert (symptom ?sym)))

```

Εισάγουμε headache:

```

headache
FIRE      5 flu4.1.2: f-10,f-6,f-2,f-14
You have flu with certainty factor 0.7 or common cold with certainty factor 0.4
Enter end if you don't have other symptoms
You might also have one of the following symptoms:
chills
sneeze
sore_throat
enter which one

```

Ο κανόνας **flu4.1.2** είναι:

```

(defrule flu4.1.2
(declare (salience 40))

(symptom fever)

(symptom cough)

(symptom tiredness)

(symptom headache)

```

=>


```
(printout t "You have flu with certainty factor 0.7 or common cold with  
certainty factor 0.4" crlf)
```

```
(printout t "Enter \"end\" if you don't have other symptoms" crlf)
```

```
(printout t "You might also have one of the following symptoms:" crlf)
```

```
(printout t
```

```
"chills
```

```
sneeze
```

```
sore_throat" crlf)
```

```
(printout t "enter which one" crlf)
```

```
(bind ?sym(read))
```

```
(assert (cf_flu 0.7))
```

```
(assert (cf_cc 0.4))
```

```
(assert (symptom ?sym)))
```

Τέλος εισάγουμε end, παρ'όλο που δεν έχουμε καταλήξει σε ένα συμπέρασμα.

```
|end  
FIRE      6 flu4.1.2e: f-10,f-6,f-2,f-14,f-17  
You have flu with certainty factor 0.7 or common cold with certainty factor 0.4  
End of session  
CLIPS> █
```

Ο κανόνας **flu4.1.2e** είναι:

```
(defrule flu4.1.2e
```

```
(declare (salience 41))
```

```
(symptom fever)
```

```
(symptom cough)
```

```
(symptom tiredness)
```

```
(symptom headache)
```

```
(symptom end)
```

=>

```
(printout t "You have flu with certainty factor 0.7 or common cold with  
certainty factor 0.4" crlf)
```

```
(printout t "End of session" crlf)
```

```
(reset))
```

Βλέπουμε ότι η εκτέλεση του προγράμματος τερματίστηκε έχοντας δύο πιθανές ασθένειες, καμία εκ των οποίων δεν έχει συντελεστή βεβαιότητας 1. Αυτό σημαίνει ότι δεδομένων των συμπτωμάτων που εισήχθησαν στη βάση γνώσης, δεν μπορεί να βγει κάποιο βέβαιο συμπέρασμα σχετικά με τις πιθανές ασθένειες. Το μόνο που μπορούμε να συμπεράνουμε είναι ότι είναι πιο πιθανό ο ασθενής να έχει γρίπη παρά κρυολόγημα. Στο κεφάλαιο 4 θα δούμε τρόπους με τους οποίους μπορούμε να συνεχίσουμε από αυτό το σημείο και να καταλήξουμε σε ασθένεια με μεγαλύτερο συντελεστή βεβαιότητας.

Κεφάλαιο 4: Μελλοντικές επεκτάσεις

Προσθήκες στο υπάρχον πρόγραμμα

Επιπλέον συντελεστές βεβαιότητας

Στην παρούσα εκδοχή του προγράμματος χρησιμοποιήσαμε συντελεστή βεβαιότητας για να δηλώσουμε το βαθμό βεβαιότητας με τον οποίο δεχόμαστε ένα συμπέρασμα, στην προκειμένη περίπτωση το βαθμό βεβαιότητας με τον οποίο υφίσταται μια ασθένεια. Στην ουσία δώσαμε συντελεστή βεβαιότητας μόνο στο αποτέλεσμα της εφαρμογής. Δεχτήκαμε σαν δεδομένο ότι η ύπαρξη κάποιου συμπτώματος προκαλεί συγκεκριμένη ασθένεια. Πως μπορούμε να είμαστε σίγουροι ότι το σύμπτωμα αυτό είναι επαρκές; Πάνω από ποιες τιμές θεωρείται ότι κάποιος έχει πυρετό; Για να απαντήσουμε σε αυτά τα ερωτήματα και για να εξάγουμε πιο ρεαλιστικά συμπεράσματα, θα μπορούσαμε να δώσουμε συντελεστή βεβαιότητας και στα συμπτώματα, δηλαδή στα γεγονότα. Ας δούμε ένα παράδειγμα:

- **AN**

{ (σύμπτωμα=πυρετός) ΚΑΙ (συντελεστής βεβαιότητας πυρετού=0.7) ΚΑΙ (σύμπτωμα=πονοκέφαλος) ΚΑΙ (συντελεστής βεβαιότητας πονοκέφαλου=0.5) }

TOTE

**{ (ασθένεια=γρίπη) (συντελεστής βεβαιότητας γρίπης=0.5)
(συντελεστής βεβαιότητας αποτελέσματος=0.25) }**

Ο συντελεστής βεβαιότητας συμπεράσματος προκύπτει από την ελάχιστη τιμή των συντελεστών βεβαιότητας της υπόθεσης επί τον συντελεστή βεβαιότητας του αποτελέσματος.

Συντελεστής βεβαιότητας υπόθεσης=MIN(συντελεστής βεβαιότητας πυρετού, συντελεστής βεβαιότητας πονοκέφαλου)

Συντελεστής βεβαιότητας αποτελέσματος=(Συντελεστής βεβαιότητας υπόθεσης)*(Συντελεστής βεβαιότητας γρίπης)

Στον κώδικα του προγράμματος CLIPS αυτό ο κανόνας θα γινόταν:

```
(defrule flu2.1.2
```

```
(declare (salience 20))
```

```
(symptom fever)
```

```
(cf_fever 0.7)
```

```
(symptom headache)
```

```
(cf_headache 0.5)
```

```
=>
```

```
(printout t "You have flu with certainty factor 0.5 " crlf)
```

```
(printout t "Enter "end" if you don't have other symptoms" crlf)
```

```
(printout t "You might also have one of the following symptoms:" crlf)
```

```
(printout t "cough,tiredness,chills,sneeze,sore_throat" crlf)
```

```
(printout t "enter which one" crlf)
```

```
(assert (cf_flu 0.5))
```

```
(assert(cf_conclusion 0.25))
```

```
(assert(disease flu))
```

```
(bind ?sym(read))
```

```
(assert (symptom ?sym)))
```

Με αυτό τον τρόπο διαχωρίζονται οι περιπτώσεις των ασθενών με ελαφρά και βαριά συμπτώματα βάσει του συντελεστή βεβαιότητας αποτελέσματος. Ένας ασθενής με ελαφρά συμπτώματα θα έχει σαφώς μικρότερο συντελεστή βεβαιότητας αποτελέσματος από κάποιον με βαριά συμπτώματα. Αυτός ο διαχωρισμός δίνει τη δυνατότητα στο γιατρό-χρήστη του προγράμματος να ακολουθήσει διαφορετική αντιμετώπιση της ασθένειας σε κάθε περίπτωση.

Επιπλέον μετρήσεις και δείκτες

Για ακόμα πιο ακριβή αποτελέσματα θα μπορούσαμε πέρα από συμπτώματα να χρησιμοποιήσουμε και άλλες τιμές όπως ηλικία, αιματοκρίτης, λευκά αιμοσφαίρια κλπ τα οποία αναλόγως με τις τιμές τους μπορούν να δώσουν διαφορετικό συντελεστή βεβαιότητας αποτελέσματος ή ακόμα και να αποκλείσουν ασθένειες.

Παράδειγμα:

```
(defrule ost4
```

```
(declare (salience 40))
```

```
(symptom bone_fracture)
```

```
(symptom loss_of_height)
```

```
(symptom curvature)
```

```
(symptom back_pain)
```

```
(age >40)
```

```
=>
```

```
(printout t "You have osteoporosis with certainty factor 1" crlf)
```

```
(printout t "Enter \"end\" " crlf)
```

```
(assert (disease osteoporosis))
```

```
(assert (cf_ost 1))
```

```
(bind ?sym(read))
```

```
(assert (symptom ?sym)))
```

Εάν απουσίαζε το **(age >40)** τότε ο κανόνας θα ήταν ψευδής παρ'όλα τα συμπτώματα καθώς ο ασθενής είναι απίθανο να παρουσιάσει οστεοπόρωση σε ηλικία μικρότερη των 40.

Εξέλιξη του προγράμματος

Εάν υποθέσουμε ότι το πρόγραμμά μας λειτουργεί με όσο το δυνατόν περισσότερη ακρίβεια και τα αποτελέσματα που παράγει είναι ρεαλιστικά, τότε βάσει αυτών των αποτελεσμάτων θα μπορούσε το πρόγραμμα να παράγει τις συνιστώμενες θεραπείες για την κάθε ασθένεια. Το πρόγραμμα θα μπορούσε να λειτουργεί ως εξής:

AN

{ (ασθένεια=...) ΚΑΙ (συντελεστής βεβαιότητας αποτελέσματος=...) ΚΑΙ (ηλικία=...) }

TOTE

{ (θεραπεία=...) }

Συγκεκριμένα, μετά την εκτέλεση του παρόντος προγράμματος, τα αποτελέσματα εισάγονται στη βάση γνώσης. Ένα δεύτερο πρόγραμμα χρησιμοποιεί αυτή τη βάση γνώσης και χρησιμοποιεί τα γεγονότα της σε κανόνες της μορφής:

```
(defrule treatment1
```

```
(declare (salience 10))
```

```
(disease flu)
```

```
(cf_conclusion 0.72)
```

```
(age <40)
```

```
=>
```

```
(printout t "The treatment is antibiotics" crlf)
```

```
(assert(treatment antibiotics)))
```

Κεφάλαιο 5: Βιβλιογραφία

- [1] http://en.wikipedia.org/wiki/Decision_support_system
- [2] http://en.wikipedia.org/wiki/Clinical_decision_support_system
- [3] <http://clipsrules.sourceforge.net/>
- [4] <http://www.ygeianet.gov.gr/treatment.aspx>
- [5] www.wikiversity.com
- [6] www.openclinical.org
- [7] <http://medlab.cs.uoi.gr>
- [8] <http://neamh.cns.uni.edu/MedInfo/mycin.html>
- [9] <http://neamh.cns.uni.edu/MedInfo/oncocin.html>
- [10] <http://inf-server.inf.uth.gr/courses/CE322/>
- [11] <http://inf-server.inf.uth.gr/courses/CE326/>

